

Predicție liniară

Lab 13, PSS

Table of contents

1	Obiectiv	1
2	Noțiuni teoretice	1
3	Exemple Matlab	2
3.1	Generarea unor semnale simple	2
3.2	Calcul coeficienți LPC	3
3.3	Predicția cu coeficienții LPC	3
4	Exerciții teoretice	3
5	Exerciții practice	3
6	Întrebări finale	5

1 Obiectiv

Studiul predicției liniare a semnalelor.

2 Noțiuni teoretice

Predicția liniară reprezintă estimarea unui eșantion al semnalului $x[n]$ ca o combinație liniară a N eșantioane precedente:

$$x[n] \approx a_1 x[n-1] + a_2 x[n-2] + \dots + a_N x[n-N]$$

Semnalele care respectă (aproximativ) o astfel de relație se numesc “autoregresive” (AR). N reprezintă ordinul modelului autoregresiv.

În Matlab, funcția `lpc()` estimează coeficienții a_k (citiți documentația).

O metodă alternativă, mai exactă, este furnizată în funcția `lpc_exact()` împreună cu lucrarea de laborator.

3 Exemple Matlab

3.1 Generarea unor semnale simple

Semnal liniar crescător:

```
x = linspace(0,20,50);  
plot(x)
```

Semnal sinusoidal:

```
n = 0:50;  
f = 0.1;  
x = sin(2*pi*f*n);  
plot(x)
```

Semnal exponential $a^n u[n]$:

```
n = 0:20;  
x = (1/2).^n;  
plot(x)
```

Semnal de tip zgomot:

```
x = randn(1,100); % or use rand()  
plot(x)
```

3.2 Calcul coeficienți LPC

```
x = 1:1:10  
order = 5;  
a = lpc(x, order);
```

3.3 Predicția cu coeficienții LPC

```
x = 1:1:500  
order = 5;  
a = lpc(x, order);  
  
% Predict value at n=11 and append to x:  
n=501;  
x(n) = sum( x(n-1:-1:n-order) .* (-a(2:end)) )
```

4 Exerciții teoretice

1. Se consideră sistemul descris de ecuația cu diferențe

$$y[n] = 0.8y[n-1] + x[n] + x[n-1],$$

unde $x[n]$ este un proces aleator staționar cu medie 0 și autocorelație $\gamma_{xx}[m] = \left(\frac{1}{2}\right)^{|m|}$

- a. Determinați densitatea spectrală de putere a ieșirii $y[n]$;
- b. Determinați funcția de autocorelație a ieșirii, $\gamma_{yy}[m]$;
- c. Determinați varianța σ_y^2 a ieșirii.

5 Exerciții practice

1. Predicție liniară pe un semnal liniar

- Generați un semnal liniar crescător de 200 eșantioane, cu pantă constantă $\Delta = 0.5$, prima valoare fiind 5.

Folosiți `linspace()` sau `start:step:stop`.

- Modelăm semnalul ca un proces autoregresiv de ordin 4, AR(4). Calculați coeficienții de predicție a_k cu funcția Matlab `lpc()`.

- Pe baza coeficienților de predicție, folosind relația de predicție, preziceți următoarele 200 eșantioane ale semnalului. Afișați întregul semnal rezultat (400 eșantioane)

Puteți folosi o relație de forma `sum(x(n-1:-1:n-ordin) .* (-a(2:end)))`

- Utilizați funcția `lpc_exact()` în locul `lpc()`. Ce se observă ?
- Generați același semnal crescător cu lungime 400 eșantioane direct cu formula inițială. Afișați pe aceeași figură semnalul acesta și semnalul precedent (2 x 400 eșantioane).

Ce calitate are porțiunea prezisă, comparativ?

- Schimbați ordinul modelului în AR(1), AR(2), AR(3), AR(10). Ce se observă ?

Care este cel mai mic ordin pentru care predicția reușește?

2. Predicție liniară pe diverse semnale.

Repetăți ex. precedent pentru un semnal de forma:

- Semnal exponențial: $x[n] = (0.9)^n u[n]$. Porniți de la un semnal de lungime 50, și estimați următoarele 50 eșantioane.
- Semnal sinusoidal: $x[n] = 3 \cdot \sin(2 * \pi * f * n) u[n]$, $f = 0.05$. Porniți de la un semnal de lungime 50, și estimați următoarele 50 eșantioane.
- Sinusoidă exponențială: $x[n] = 0.8^n \cdot \sin(2 * \pi * f * n) u[n]$, $f = 0.2$. Lungime 50 + 50
- Semnal sinusoidal atenuat: $x[n] = \frac{\sin(2 * \pi * f * n)}{2 * \pi * f * n} u[n]$, $f = 0.05$. Porniți de la un semnal de lungime 50, și estimați următoarele 50 eșantioane.
- Semnal de tip zgomot alb gaussian (AWGN, generat cu `randn()`). $x[n] = 0.8^n \cdot \sin(2 * \pi * f * n) u[n]$, $f = 0.2$. Lungime 500 + 100. Apoi lungime 20 + 100.
- Semnal de tip zgomot alb uniform (generat cu `rand()`). $x[n] = 0.8^n \cdot \sin(2 * \pi * f * n) u[n]$, $f = 0.2$. Lungime 500 + 100. Apoi lungime 20 + 100.
- Semnal sinusoidal in zgomot alb: $x[n] = 2 \cdot \sin(2 * \pi * f * n) u[n] + AWGN$, $f = 0.05$. Lungime 100 + 100.
- Semnalul `mtlb` încărcat cu `load mtlb;`. Estimați următoarea secundă de semnal audio.
- Primele 150 de eșantioane din semnalul `mtlb`. Estimați următoarea secundă de semnal audio.

3. Reducerea zgomotului prin predicție.

Generați un semnal de forma:

$$x[n] = \sin(2 * \pi * f * n) + \text{zgomot alb.}$$

Calculați coeficienții de predicție, și apoi estimați fiecare eșantion din semnalul $x[n]$ pe baza eșantioanelor precedente. Afișați semnalul astfel obținut ($x_2[n]$) cu semnalul original pe aceeași figură. Ce se observă?

4. Detecția vocii (Voice Activity Detector).

- Încărcați semnalul audio `data_slow.wav` (cu `audioread()`), afișați-l grafic și redați-l audio.
 - a. Utilizați funcția `buffer()` pentru a împărți semnalul în ferestre cu lungimea de aproximativ 25ms.
 - b. Modelați fiecare segment semnalul ca un proces aleator AR(12), și găsiți coeficienții liniari de predicție pentru fiecare segment.
 - c. Pentru fiecare segment, calculați energia coeficienților de predicție (suma coeficienților la pătrat). Afișați secvența de valori obținută.
 - d. De pe grafic, alegeți un prag convenabil pentru a diferenția segmentele de voce de cele de pauză.
 - e. Eliminați segmentele din semnal care sunt de pauză, și reuniți segmentele rămase într-un semnal întreg. Ascultați semnalul astfel obținut.

5. Repetați exercițiul anterior, dar adăugând peste semnalul inițial zgomot AWGN. Până la ce nivel de zgomot se obțin rezultate bune?

6. Incărcați imaginea `lena512.bmp`. Transformați în 0 valorile de pe liniile 20 : 30, coloanele de la 100 la 150.

Refaceți imaginea în felul următor: pentru fiecare linie separat, modelați primele 100 eșantioane cu un proces AR(10), apoi estimați cele 50 valori lipsa care urmează.

Afișați imaginea astfel obținută.

6 Întrebări finale

1. TBD